Lecture 1:
Introduction to Digital Logic Design

CSE 140: Components and Design Techniques for Digital Systems
Winter 2016

CK Cheng
Dept. of Computer Science and Engineering
University of California, San Diego
Information about the Instructor

• Instructor: CK Cheng
• Education: Ph.D. in EECS UC Berkeley
• Industrial Experiences: Engineer of AMD, Mentor Graphics, Bellcore; Consultant for technology companies
• Email: ckcheng+140@ucsd.edu
• Office: 2130 EBU3B
• Office hours will be posted on the course website
• Websites
  – http://cseweb.ucsd.edu/~kuan
  – http://cseweb.ucsd.edu/classes/wi16/cse140-a
Information about TAs

TAs
• Ardeshiricham, Armaiti, aardeshi@ucsd.edu
• Hingolikar, Mrinmayee Pravin, mHINGoli@ucsd.edu
• Kang, Ilgweon, iikang@ucsd.edu
• Maurya, Akanksha, akmaurya@ucsd.edu
• Wang, Xinyuan, xiw193@ucsd.edu

Tutors
• Fakhourian, Eric, efakhour@ucsd.edu
• Shih, Linda, lishih@ucsd.edu
• Wang, Runping, ruw042@ucsd.edu

Office hours will be posted on the course website
Logistics: Resources

All information about the class is on the class website: http://cseweb.ucsd.edu/classes/wi16/cse140-a/index.html

- Approx. Syllabus
- Detailed schedule
- Readings
- Assignments (Piazza)
- Grading policy (Website)
- Forum (Piazza)

- Content/announcements and grades will be posted through Piazza *make sure you have access

I will assume that you check these daily. Office hours and emails will be available on the course website.
Logistics: Textbooks

Required text:

• Online Textbook: Digital Design by F. Vahid
  1. Sign up at zyBooks.com
  2. Enter zyBook code UCSDCSE140Winter2016
  3. Click Subscribe

Reference texts (recommended and reserved in library)

• Digital Systems and Hardware/Firmware Algorithms, Milos D. Ercegovac and Tomas Lang
Lecture: Peer Instruction

• I will pose questions. You will
  – Solo vote: Think for yourself and select answer
  – Discuss: Analyze problem in teams of three
    • Practice analyzing, talking about challenging concepts
    • Reach consensus
  – Class wide discussion:
    • Led by YOU (students) – tell us what you talked about in discussion that everyone should know.

• Many questions are open, i.e. no exact solutions.
  – Emphasis is on reasoning and team discussion
  – No solution will be posted
Logistics: Grading

Grade on style, completeness and correctness

- zyBook exercises: 10%
- iClicker: $x=5\%$ (by participation up to 20 classes)
- Homework: 10-$x\%$ (grade based on a subset of problems. If more than 70\% of class fill out CAPE evaluations, the lowest homework score will be dropped)
- Midterm 1: 25\% (M 1/25)
- Midterm 2: 25\% (W 2/17)
- Final: 30\% (3-5PM, M 3/14)
- Grading: The best of the following
  - The absolute: A- $>90\%$; B- $>80\%$ of total 100\% score
  - The curve: (A+,A,A-) top $33+\varepsilon\%$ of class; (B+,B,B-) second $33+\varepsilon\%$
  - The bottom: C- above 45\% of absolute score.
A word on the grading components

• Exercises:
  – Interactive learning experience

• iClicker:
  – Clarification of the concepts and team discussion

• HWs:
  – Practice for exams. Group discussion is encouraged
  – However, we are required to write them individually for best results

• Exams
  – (Another) Indication of how well we have absorbed the material
  – Solution and grading policy will be posted after exam.
  – Learn from mistakes and move on ….
Course Problems…Cheating

• What is cheating?
  – *Studying* together in groups is *encouraged*
  – Turned-in work must be *completely* your own.
  – Copying someone else’s solution on a HW or exam is cheating
  – Both “giver” and “receiver” are equally culpable

• We have to address the issue once the cheating is reported by TAs or tutors.
Motivation

• Microelectronic technologies have revolutionized our world: cell phones, internet, rapid advances in medicine, etc.

• The semiconductor industry has grown from $21 billion in 1985 to $315 billion in 2013.
The Digital Revolution

Integrated Circuit: Many digital operations on the same material

Vacuum tubes

Exponential Growth of Computation

ENIAC

World War II

Stored Program Model

1949

1965

Moore’s Law

Integrated Circuit

(1.6 x 11.1 mm)
Building complex circuits

Transistor

<table>
<thead>
<tr>
<th>B</th>
<th>A</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Robert Noyce, 1927 - 1990

- Nicknamed “Mayor of Silicon Valley”
- Cofounded Fairchild Semiconductor in 1957
- Cofounded Intel in 1968
- Co-invented the integrated circuit
Gordon Moore

• Cofounded Intel in 1968 with Robert Noyce.
• Moore’s Law: the number of transistors on a computer chip doubles every 1.5 years (observed in 1965)
Technology Trends: Moore’s Law

- Since 1975, transistor counts have doubled every two years.
How do we handle complexity?

- Big idea: Coordination of many levels of abstraction

CSE 140

Software

Hardware

CSE 140,141

Instruction Set Architecture

Datapath & Control

Digital Design

Circuit Design

Transistors

CSE 120

Algos: CSE 100, 101

CSE 140

Application (ex: browser)

Compiler

Assembler

Operating System (Mac OSX)
Principle of Abstraction

Abstraction: Hiding details when they are not important
Scope

• The purpose of this course is that we:
  – Learn the principles of digital design
  – Learn to systematically debug increasingly complex designs
  – Design and build digital systems
  – Learn what’s under the hood of an electronic component
Scope: Overall Picture of CS140

Data Path Subsystem

- Memory File
- Pointer
- ALU
- Memory Register
- Conditions

Control Subsystem

- Conditions
- Sequential machine
- Control
- CLK: Synchronizing Clock
Combinational Logic vs Sequential Network

**Combinational logic:**

\[ y_i = f_i(x_1, \ldots, x_n) \]

**Sequential Networks**

1. Memory
2. Time Steps (Clock)

\[
\begin{align*}
  y_i^{t+1} &= f_i(x_1^t, \ldots, x_n^t, s_1^t, \ldots, s_m^t) \\
  s_i^{t+1} &= g_i(x_1^t, \ldots, x_n^t, s_1^t, \ldots, s_m^t)
\end{align*}
\]
## Scope

<table>
<thead>
<tr>
<th>Subjects</th>
<th>Building Blocks</th>
<th>Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combinational Logic</td>
<td>AND, OR, NOT, XOR</td>
<td>Boolean Algebra</td>
</tr>
<tr>
<td>Sequential Network</td>
<td>AND, OR, NOT, FF</td>
<td>Finite State Machine</td>
</tr>
<tr>
<td>Standard Modules</td>
<td>Operators, Interconnects, Memory</td>
<td>Arithmetics, Universal Logic</td>
</tr>
<tr>
<td>System Design</td>
<td>Data Paths, Control Paths</td>
<td>Methodologies</td>
</tr>
</tbody>
</table>
Combinational Logic Basics
What is a combinational circuit?

- No memory
- Realizes one or more functions
- Inputs and outputs can only have two discrete values
  - Physical domain (usually, voltages) (0V, 5V)
  - Mathematical domain: Boolean variables (true or false)

Differentiate between different representations:
- physical circuit
- schematic diagram
- mathematical expressions
Representations of combinational circuits: The Schematic

• What is the simplest combinational circuit that you know?
Representations of combinational circuits:

Truth Table

<table>
<thead>
<tr>
<th>AND</th>
<th>A B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0 1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1 0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1 1</td>
<td>1</td>
</tr>
</tbody>
</table>

[Diagram of an AND gate]
Representations of combinational circuits:

Boolean Expression/Equation

\[ Y = AB \]

All three forms are equivalent!
Boolean Algebra

Similar to regular algebra but defined on sets with only three basic ‘logic’ operations:

1. Intersection: AND (2-input); Operator: .
2. Union: OR (2-input); Operator: +
3. Complement: NOT (1-input); Operator: ‘
Boolean algebra and switching functions

Two-input AND (\(\cdot\))

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

For an AND gate,
0 at input blocks the other inputs and dominates the output
1 at input passes signal A

Two-input OR (+)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

For an OR gate,
1 at input blocks the other inputs and dominates the output
0 at input passes signal A

One-input NOT (Complement, ')'

<table>
<thead>
<tr>
<th>A</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

A
Boolean Algebra

iClicker Q: For two Boolean variables $X$ and $Y$ with $X=1$, $Y=0$, what is function $F(X,Y)=X+Y$?

A. $F(X,Y)=0$

B. $F(X,Y)=1$

C. $F(X,Y)=2$
iClicker Q: For two Boolean variables X and Y with X=1, Y=0, what is function $F(X,Y)=X+X+Y$?

A. $F(X,Y)=0$

B. $F(X,Y)=1$

C. $F(X,Y)=2$
Boolean Algebra

iClicker Q: For two Boolean variables X and Y with X=1, Y=0, what is function $F(X,Y)=X+XY$?

A. $F(X,Y)=0$
B. $F(X,Y)=1$
C. $F(X,Y)=2$
Boolean Algebra

iClicker Q: For two Boolean variables $X$ and $Y$ with $X=1$, $Y=0$, what is function $F(X,Y)=(X+Y)Y$?

A. $F(X,Y)=0$
B. $F(X,Y)=1$
C. $F(X,Y)=2$
Boolean Algebra

Boolean operations satisfy the following laws:

• Commutative laws: $a + b = b + a$, $a \cdot b = b \cdot a$
• Distributive laws: $a + (b \cdot c) = (a + b) \cdot (a + c)$,
  $a \cdot (b + c) = a \cdot b + a \cdot c$
• Identity laws: $a + 0 = a$, $a \cdot 1 = a$
• Complement laws: $a + a' = 1$, $a \cdot a' = 0$
So, what is the point of representing gates as symbols and Boolean expressions?

• Given the Boolean expression, we can draw the circuit it represents by cascading gates (and vice versa)

Logic circuit vs. Boolean Algebra Expression
Next class

• Designing Combinational circuits